# IOT Datacommunicatie Project verslag

Long Range Keyboard

Door Ruben Schoonbaert

**Projectvoorstel:**

De bedoeling van dit voorstel is voor het maken van een toetsenbord dat draadloos kan communiceren met een laptop via USB op een lange afstand. Het toetsenbord zal een oud toetsenbord zijn met een PS/2 connector. Omdat het PS/2 dataprotocol gemakkelijk kan worden verbonden met een microcontroller zonder de nood voor extra hardware.

*Fig. PS/2 Connector*

De microcontroller zal een raspberry pi pico zijn (geprogrammeerd via Python). Deze neemt het signaal van het toetsenbord en stuurt een het via SPI naar een nrf24l01+ RF module. Deze module zal de data draadloos versturen. (2.4GHz band).

De ontvanger zal een nrf24l01 module zijn. Deze is verbonden via SPI met een Digispark Attiny85 bord. Ik heb voor dit bord gekozen omdat het zich als een HUID (Human interface device) kan voordoen. En het kan dus rechtstreeks USB toestellen namaken. (Iets wat de Serial chip van een Arduino dit normaal moeilijk maakt).

In een ideale setting zou dit toetsenbord input kunnen geven voor een computer met een bereik van 800m (volgens de datasheet) maar ik verwacht dat dit in praktijk een stuk minder zal zijn.

Afbeelding met diagram

Automatisch gegenereerde beschrijving**Blokschema:**

**Uitwerking Project: Transmitter**

Afbeelding met tekst, elektronica

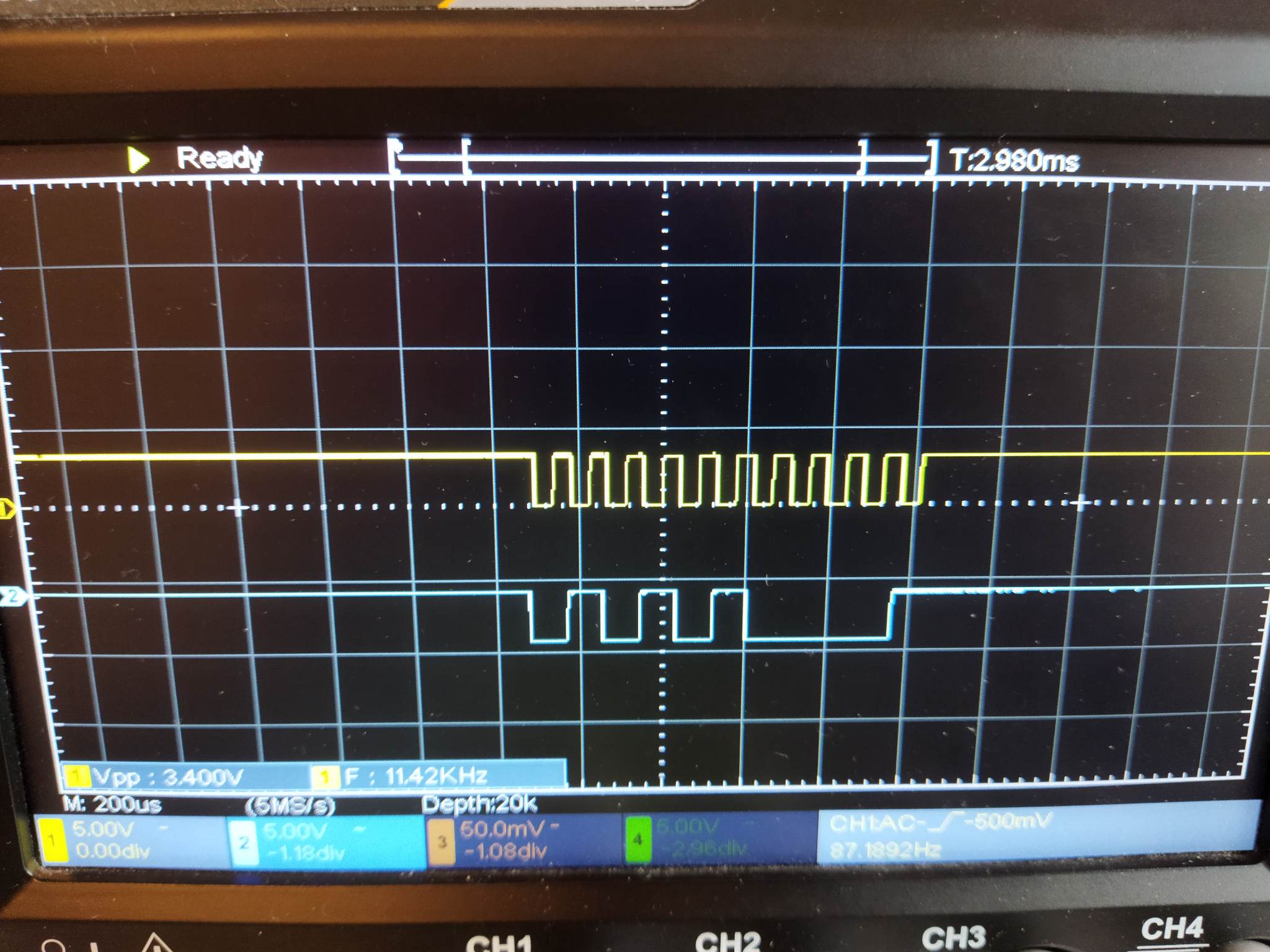
Automatisch gegenereerde beschrijving

De hardware van de transmitter module bestaat uit een Raspberry Pi Pico als microcontroller, Een NRF24L01+ module als radio module en een BI-Directionele logic shifter gebaseerd op de BSS138 MOSFET om de 5v logica om te zetten naar 3.3v. Er is ook een 10uF elektrolytische condensator geplaatst over VBus en GND om eventuele spanningspieken voor het toetsenbord te vermijden.

Afbeelding met diagram, schematisch

Automatisch gegenereerde beschrijving

Loop ontwikkeling transmitter

Het eerste waaraan ik begon was het uitmeten van de PS/2 Connector om te bepalen welke functie de kabels hadden en om een kijk te nemen naar het dataprotocol zodat ik meer wist waar ik mee bezig was.

**Data**

**CLK**

Het PS/2 protocol is een gelijkaardig protocol als RS-232 met 1 start bit, 8 data bits, 1 parity bit en 1 stop bit. Verder teste ik het het toetsenbord via de PS2Keyboard Library op een Arduino Uno. Hier kwam ik tot de conclusie dat ik de data uit het toetsenbord ietswat vervormd was en niet altijd correct functioneerde als het niet gevoed werdt met 5v. Vandaar de toevoeging van een BI directionele level shifter.

Origineel was ik van plan om de Raspberry pi pico te programmeren met Python. En ik heb een tijd bezig geweest met een werkend PS/2 protocol omvormer te maken in Python. Maar mijn beknopte kennis van Python en de gelimiteerde ondersteuning voor de Raspberry pi pico (De Pico is nog maar van Januarie 2021 op de markt dus de documentatie en troubleshooting is minder aanwezig dan traditionele microcontrollers). Moest ik dit achterwege laten en de microcontroller programmeren met C++.

Ik heb toen de PS2Keyboard.h library wat moeten aanpassen tegen dat het werkte met de Raspberry Pi Pico.

Hiernaa voegde ik de NRF24L01+ module toe. Deze wordt gestuurd door SPI. Op de default bus (SPI0). Wanneer een karakter wordt opgevangen door het toetsenbordt wordt ze omgezet naar een character. Deze wordt verstuurd door De NRF24L01+ module op het kanaal “00001”.

Afbeelding met tekst, elektronica

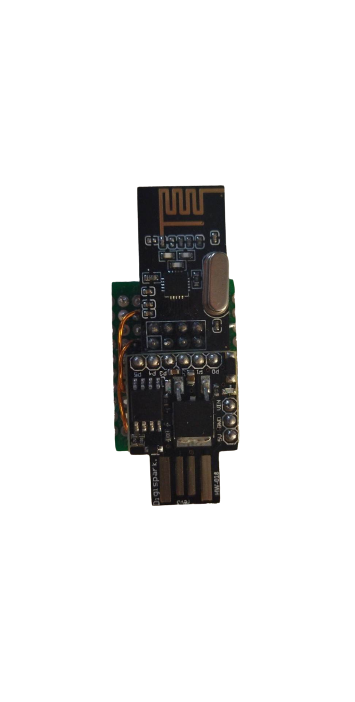
Automatisch gegenereerde beschrijving**Uitwerking Project: Reciever**

De hardware van de reciever bestaat uit een Digispark ATTINY85. Een Arduino nano en een NRF24L01 radio module. De Arduino nano is verbonden met de RadioModule via SPI en vangt inkomede characters om. Hij zet deze om naar ASCII key codes voor de Digispark en stuurt deze naar de Digispark via I2C. De digispark gedraagd zich als een HID en stuurt deze ASCII codes naar de computer met behulp van de DigiKeyboardFr.h library.

Heel de reciever wordt gevoegdt door de 5V van de USB poort.

Afbeelding met diagram, schematisch

Automatisch gegenereerde beschrijving

Loop ontwikkeling reciever.

Ik begon met het samenstellen van de reciever. Volgens de documentatie van de Digispark heeft de ATTINY85 een ingebouwde SPI poort. Dus soldeerde ik deze uiteraart aan de NRF240L1 module.

Nu het blijkt dat alhoewel de ATiny85 een SPI bus heeft, de library dat Digistump aanbied om te communiceren over de SPI bus niet werkt. Het is een verouderde versie van de normale SPI.h library dat ze nooit hebben aangepast om te werken voor de ATiny85. En alhoewel er een aantal reparaties en technieken bestaan kreeg ik geen enkele robuust werkende op dit systeem dus moest ik werken met nog een extra microcontroller.

Vandaar dat ik een Arduino Nano er tussen heb gestoken. De arduino Nano krijgt dus de karakters binnen van de NRF24L01 module binnen en zet deze om naar een ASCII code aan de hand van een keymap. En verstuurt ze naar de Digispark.

Initeel was ik vanplan om Seriële communicatie te gebruiken tussen de Digispark en de Arduino Nano maar ik ondervinde dat de Attiny85 maar 1 programmable interupt heeft en SoftwareSerial.h (Library voor seriële communicatie) en DigiSparkKeyboard.h (HID keyboard library).

Om dit op te lossen schakelde ik over naar I2C voor de data tussen de microcontrollers te sturen. Met adress 0x08.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

De digispark luistert naar dit adress en als het een code binnenkrijgt met TinyWireS.h (library voor digispark I2C communicatie) Op dat adress stuurt hij deze code naar de computer via DigikeyboardFr.h

Op het moment heb ik alleen alle letters en spatie, enter & backspace toegevoegd. De rest zou zonder veel moeite moeten kunnen worden toegoevoegd maar dit leek me genoeg om als test principe te tonen.

Code:

Alle code kan teruggevonden worden op deze repository:

*https://github.com/RubenSchoonbaert/LongRangeKeyboard*